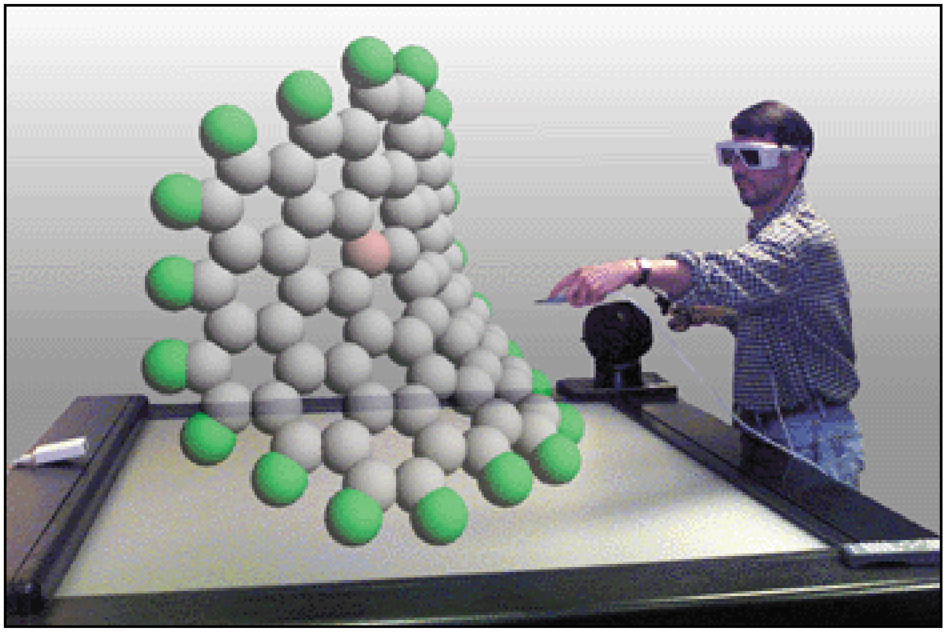
NASnews
March – April 1999 • Volume 4, Number 2

In this issue...

Steering Molecules by Feel



The Virtual Mechanosynthesis (VMS) utility allows users to see, move, and even "feel" simulated molecular structures in three dimensions. In the graphite sheet being explored by VMS designer Chris Henze, above, gray spheres represent carbon atoms and green spheres represent hydrogen. Using the pointer, Henze has "grabbed" the pink atom for manipulation. *Photo by Michael Boswell. Dataset by Chris Henze.*

To The Article...

No one doubts that everything around us is composed of atoms, but there is still something unsatisfyingly abstract about the atomic theory of matter. Since atoms cannot be observed directly, we can see them only in our imagination. The images from our most powerful microscopes are barely suggestive of the intricate and orderly atomic arrangements in nature, and atomic motions and interactions are far too rapid for us to follow directly.

Recently, however, another way of visualizing atoms has become available. By coupling an accurate molecular dynamics simulation code to an immersive graphical display with interactive capabilities and manual force feedback, researchers in the NAS data analysis group have provided users with an opportunity to interact (virtually) with small collections of atoms "first hand."

To The Article...

NASnews

▶ Main Menu

March – April 1999 • Volume 4, Number 2

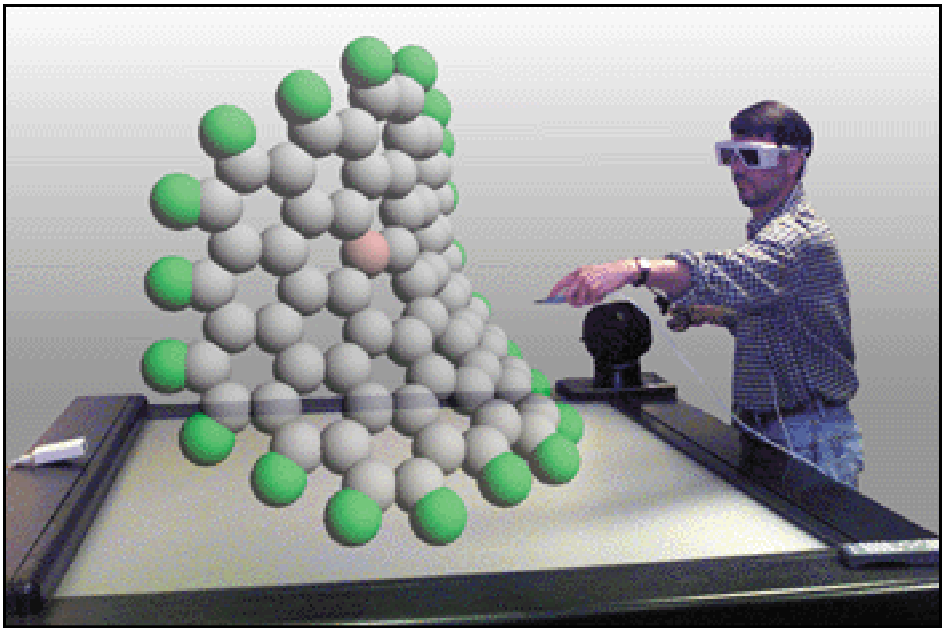
Steering Molecules by Feel

by Chris Henze

No one doubts that everything around us is composed of atoms, but there is still something unsatisfyingly abstract about the atomic theory of matter. Since atoms cannot be observed directly, we can see them only in our imaginations. The images from our most powerful microscopes are barely suggestive of the intricate and orderly atomic arrangements in nature, and atomic motions and interactions are far too rapid for us to follow directly.

Recently, however, another way of visualizing atoms has become available. By coupling an accurate molecular dynamics simulation code to an immersive graphical display with interactive capabilities and manual force feedback, researchers in the NAS data analysis group have provided users with an opportunity to interact (virtually) with small collections of atoms "first hand."

A user of the application, called "Virtual Mechanosynthesis" or VMS, sees various collections of atoms floating in space above the NAS Visualization Laboratory's Immersive Workbench, made by Fakespace Inc. Like tinkertoys, the atoms can be moved about and built into arbitrarily complex structures -- a form of assembly that is the essence of nanotechnology. Much effort in current nanotechnology is directed at finding plausible designs for nanoassemblers, mechanisms capable of placing individual atoms in precisely defined positions (mechanosynthesis). Functioning nanoassemblers will require atomic blueprints and programs to guide their actions. By allowing researchers to explore, rehearse, and debug complex assembly sequences, the VMS will help in the creation of plausible atomic designs.



The Virtual Mechanosynthesis (VMS) utility allows users to see, move, and even "feel" simulated molecular structures in three dimensions. In the graphite sheet being explored by VMS designer Chris Henze, above, gray spheres represent carbon atoms and green spheres represent hydrogen. Using the pointer, Henze has "grabbed" the pink atom for manipulation. *Photo by Michael Boswell. Dataset by Chris Henze.*

Scaling Up Space, Slowing Down Time

In VMS, which was created over the last year and demonstrated at the SC98 High Performance Networking and Computing Conference in November, atoms are represented as softball-sized spheres. Typically they are seen vibrating and rotating about their equilibrium positions. Since real atoms are about an angstrom across and vibrate at frequencies of about a Terahertz, VMS effectively scales up space by nine orders of magnitude, while slowing down time by 15 orders of magnitude.

These effects are accomplished by using the standard immersive or virtual-reality props, beginning with the Immersive Workbench, a rear-projection stereo display system. Stereo glasses are fitted with a tracking device so that head movements result in a changing viewpoint, just as if the user were peering around a real object hovering above the Workbench. The position of a handheld wand in real space above the Workbench controls a corresponding point moving about in the atomic scene. The user can use the wand to "grab space" and reposition it, in order to explore revealing views of the atomic scene.

The wand also allows the user to grab an individual atom at the touch of a button and move it about the scene at will. The ongoing simulation takes into account the changing position of the user-guided atom and sees to it that the other atoms in the scene respond in a physically realistic way.

A haptic, or force-feedback, device can also be used to interact with the simulation. This device is essentially a mouse mounted on a mechanical arm. It can be pushed around in three dimensions and, by way of a clever arrangement of small motors, it can push back. Once users "attach" to an atom using the haptic interface, they can feel the attractive and repulsive forces as this atom pushes and pulls on its neighbors. In addition to providing a convincing representation of real atoms and their interactions, the haptic interface allows users to navigate complex "energy landscapes" and thereby find energetically preferable atomic arrangements. Atoms must be nudged into appropriate positions in order to form bonds. Misplaced atoms will usually remain unattached and may even be flung aside by repulsive forces.

Hands-on Chemistry

In addition to its obvious applications in nanotechnology, VMS could be used to teach certain aspects of chemistry. Instead of using beakers of reactants at a lab bench, students could design and execute experiments involving individual atoms and molecules. General chemical principles can also be investigated. For example, if the user grabs an atom and bumps it into other atoms in a molecule, before long the whole collection of atoms will shake wildly and fly apart. The user's movements continually introduce energy into the system, which heats it up. (To accommodate this inescapable fact of physical chemistry, the VMS has a simulated thermostat to drain away excess heat; clearly, an actual nanoassembler will need a cooling system also.)

Incorporating the haptic feedback into the simulation loop presented the project's major technical challenge. Just as an animation on a computer screen must be refreshed many times per second to give the appearance of smooth movement, the force delivered by the haptic device needs to be refreshed at least 300 times per second, or the user will experience a jerkiness or buzzing sensation. However, updates from the molecular simulation are not available at such a high rate. To provide the instant gratification demanded by the virtual reality paradigm, while still avoiding the classic oscillatory behavior in feedback systems with mismatched characteristic times, we used several independent threads of control and indirectly coupled the haptic inputs and outputs to the simulation. Using a system of masses and springs, we constructed a tunable low-pass filter between the user interface and the molecular dynamics simulation.

VMS was created by researchers Chris Henze and Bryan Green in the NAS data analysis group and Creon Levit in the information technology modeling and simulation group. The molecular dynamics code incorporated in VMS uses a semi-empirical reactive bond order potential for hydrocarbons developed by Don Brenner at North Carolina State University's Material Science and Engineering Department.

VMS provides a rare example of genuine computational steering -- the ability to design and modify simulations interactively as they are running, as opposed to relegating visualization to a post-processing phase. This closed-loop paradigm allows users to immediately see the results of interactively changed parameters, providing opportunities to detect and explore patterns of cause and effect. Runtime steering is a logical evolutionary development of computational simulations, since steerable simulations allow one to conduct realistic experiments in virtual space. As a compelling example of computational steering, complete with haptic interface, the Virtual Mechanosynthesis utility effectively allows users to get a concrete feel for abstract molecular dynamics.

For more information about this research, contact Chris Henze at chenze@nas.nasa.gov.

Updated: August 3, 1999
WebWork: *Publications Media Group*
NASA Responsible Official: *Bill Feiereisen*

QUESTIONS HELP

NASnews
▶ Main Menu
March – April 1999 • Volume 4, Number 2

Experiment Shows NASA Application Can Run on IPG

by Wade Roush

Until recently, it would have been cumbersome to assign two supercomputers half a continent apart to work in parallel on a large aerodynamic simulation. No easy way existed to start a computation simultaneously on widely separated machines, or even to manage the volumes of data that must move back and forth between processors in a parallel computation. Last fall, however, members of the NAS information technology modeling and simulation group (ITMS) combined several new techniques to show, for the first time, that a tightly coupled computational fluid dynamics (CFD) code can be executed on two geographically distant machines without sacrificing performance.

Their experiments, reported at a parallel computing conference in February, also mark the first time such an application has been managed by Globus, the metacomputing toolkit developed by researchers at Argonne National Laboratory in Illinois and the Information Sciences Institute (ISI) at the University of Southern California. The group concludes in a paper delivered at the conference that "some important NASA scientific applications can be effectively implemented on the Information Power Grid" and that Globus is "a realistic starting point" for the IPG. Running large simulations efficiently on distributed computers is one important goal of IPG, NASA's participation in the national effort to build a communications infrastructure linking high-performance computers, scientific instruments, storage devices, and visualization systems (see related article, this issue). And the techniques tested by the ITMS group did just that. Using OVERFLOW^{D2}, a modified version of the well-known OVERFLOW CFD code, the group ran a simulation of an experimental aerospace vehicle on two supercomputers in California and Illinois. Their results showed that computations, though currently not competitive in speed, can be performed almost as fast as on a single machine at the NAS Facility once the proper computer support is in place.

One Step at a Time

OVERFLOW^{D2} simulates air flow around an aerospace vehicle by dividing the surrounding space into hundreds of overlapping "overset grids" of varying sizes and shapes. At each timestep in a simulation, the program computes the flow variables at each grid point. For points on grid boundaries, the program waits for updates from neighboring grids before computing a solution and moving on to the next timestep.

In OVERFLOW^{D2}, several grids are assigned to each processor in a parallel computer. If two adjoining grids are assigned to the same processor, updates can be exchanged between them almost instantly. If they are assigned to different processors, however, the data must make the trip in the form of an MPI (Message Passing Interface) message. And while the communication time for an MPI message within a single computer is usually small, this time lengthens dramatically if the message must travel to a processor in a distant machine.

The members of the ITMS team, led by Subhash Saini, set out last summer to show that a single OVERFLOW simulation could be carried out on widely separated computers using the Globus toolkit. They knew, however, that they would have to find a way to prevent lengthy communication times from causing bottlenecks in the computation. Their solution was to employ a technique called "deferred boundary update." While this method had been used before within parallel machines, no one had tried to use it to hide cross-country communication delays.

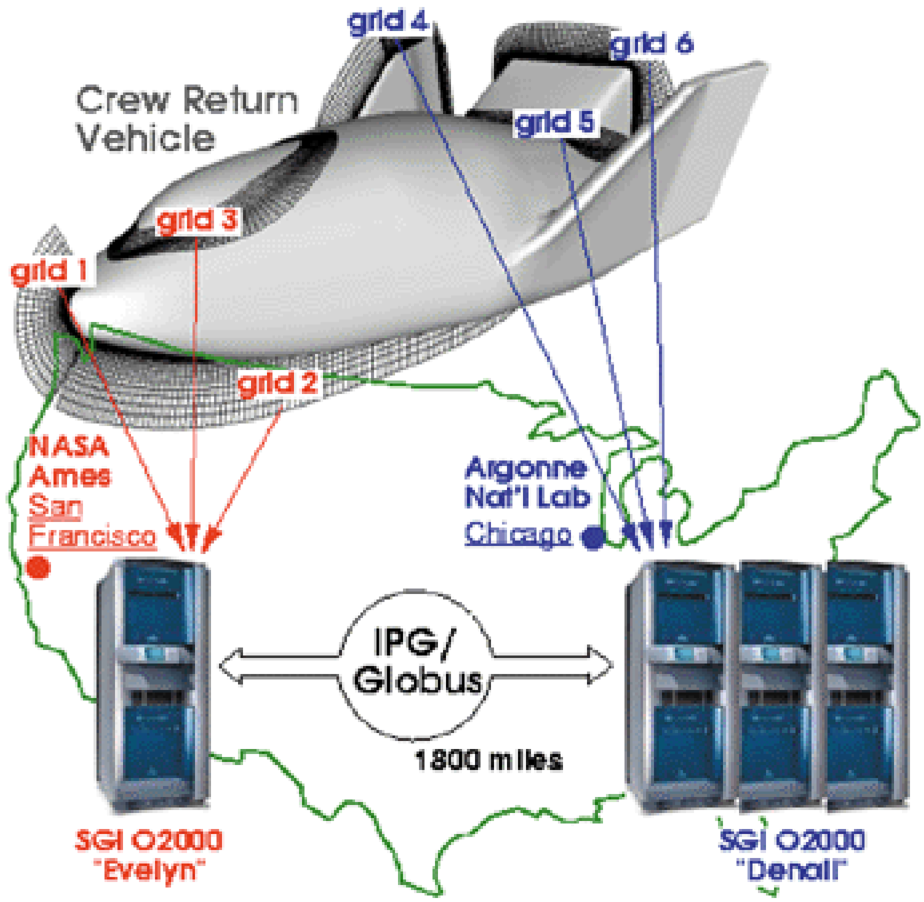
In their presentation in February at the 7th Symposium on the Frontiers of Massively Parallel Computation in Maryland, ITMS group members Saini, Stephen Barnard, Rupak Biswas, Rob Van der Wijngaart, and Maurice Yarrow, along with colleagues at Ames Research Center and Argonne, reported that an OVERFLOW simulation performed jointly by

Silicon Graphics Origin2000 computers at Argonne and the NAS Facility ran just as accurately using the deferred method as it did using the unmodified code.

Stretching the Limits

In previous experiments, loosely coupled simulations have been run on different but nearby machines, and flow solutions have been generated on one machine and visualized on another. "We wanted to stretch the limits by running a single code on different systems," says Van der Wijngaart.

The Globus team, led by Ian Foster of Argonne and Carl Kesselman of ISI, designed their toolkit with just such limit-stretching applications in mind. The collection of commands allows a user to find out which machines belonging to the Globus testbed network, spanning eight nations, match requirements for a given computational job. Then the commands help the user harness those computers together into a single "virtual machine" that can quickly run the job (see illustration, below). The Globus toolkit includes a "Grid-enabled" version of MPI, the software that coordinates communication between processors in most parallel computer programs. Saini contacted the Globus researchers, and a collaboration was born.



Graphic by Rob Van der Wijngaart

Aeromechanics Branch Contributes Code

Meanwhile, researchers Bob Meakin and Andrew Wissink in the Aeromechanics Branch of the Army/NASA Rotorcraft Division at Ames had already developed OVERFLOW^{D2}, their own parallel version of OVERFLOW using MPI. The modified code ran successfully on an IBM SP computer at the Army Corps of Engineers Waterways Experiment Station in Vicksburg, Mississippi, Meakin and Wissink found. They shared both their code and a test case simulating NASA's experimental X-38 Crew Return Vehicle with the ITMS team. It remained for the group to adapt OVERFLOW^{D2} to the Origin2000 computers at NAS and Argonne, and to determine the minimum changes needed to make the program run over a network using Globus. For the IPG to be useful in aerospace engineering and other fields, notes Yarrow, "It's important for researchers not to have to make extreme changes in their code. Otherwise they're not going to want to do it."

The three Origin2000s the ITMS group used in its experiment -- two at NAS called Evelyn and Piglet and a third at Argonne called Denali -- are separated by almost 2,000 miles. "That's the kind of sizeable distance that may be spanned on the envisioned IPG," notes Van der Wijngaart. Once the NAS researchers got OVERFLOW^{D2} running on the Origin2000s, their challenge was to make the program "latency tolerant," that is, to maintain its speed despite communication delays.

Buying Additional Speed

The researchers soon realized they could buy added speed by sacrificing absolute precision. In the original version of OVERFLOW^{D2}, boundary data at timestep n is communicated among all processors, and then the solution at timestep n+1 is computed. In the deferred scheme, the computation does not wait for the boundary data exchange to be completed. Instead, the solution at timestep n+1 is computed based in part on grid boundary data from timestep n-1. The researchers theorized that this technique would allow computation and communication to proceed simultaneously, thereby hiding the long latencies.

Although the solution from timestep n-1 is slightly different from the solution at timestep n, which introduces some error, the ITMS group believed the decrease in accuracy would be acceptably small and that the gain in speed would be significant. While this approach would not be applicable to all classes of CFD simulations, current-generation codes could probably use the technique to advantage, they thought. These were the hypotheses they set out to test in the experiments reported in the Frontiers paper.

First they applied the original and deferred versions of OVERFLOW^{D2} to a system of 128 grids from the X-38 test case, which represented the experimental Crew Return Vehicle reentering the atmosphere at a 15-degree angle of attack and a speed of Mach 1.5. For one grid near the nose cone, where the shock wave maintained a steady shape, there was no discernible difference between the solutions produced by the two methods. For another grid near the tail where the flow was unsteady, the solutions were not identical, but they both fluctuated in the same range. This confirmed that "the deferred data is almost as good [as the original version] for design purposes," says Biswas.

Cross-country Computation

Next, the group ran several simulations of the full X-38, each involving eight processors. In the first two simulations, the work was distributed across four processors in Evelyn and four processors in Piglet, with the two neighboring machines connected by a high-performance HIPPI channel. When the original version of OVERFLOW[D2] was used on this configuration, the average wall-clock time needed to compute each timestep was 16.2 seconds. When the deferred version was used, the time decreased to 12.7 seconds. While the communication time in both cases was artificially high due to the version of MPI used, the comparison proved that this method reduces communications delays significantly.

In another pair of simulations, the NAS researchers tested a version of Globus equipped with MPICH-G (the Grid-enabled version of MPI) by distributing the work across four processors in Piglet and four in Denali. Results confirmed the researchers' expectations of no improvement in speed. The reason: The DS3 Internet connection between Piglet and Denali, unlike the HIPPI connection between Piglet and Evelyn, did not allow messages to be passed asynchronously as the deferred method requires. Moreover, the DS3 connection was significantly slower than the HIPPI channel, causing both the original and deferred versions of OVERFLOW to run at the same relatively slow pace of about 21.4 seconds per timestep. But because communication time is still less than the net computational time required per timestep, "The deferred method holds the promise of hiding virtually the entire communication time," says Biswas.

The ITMS team plans further work to allow asynchronous communication between the sites, but feels the current experiment yielded helpful results. "Our test was useful exactly because it pointed to a problem that anybody doing our types of computations on the IPG is going to face," says Biswas. One reason for the limitation on asynchronous communication-the need for processing power to assist with communication-may be overcome using a method already being developed by the group and their Argonne collaborators to define "spare" processors as dedicated communications processors.

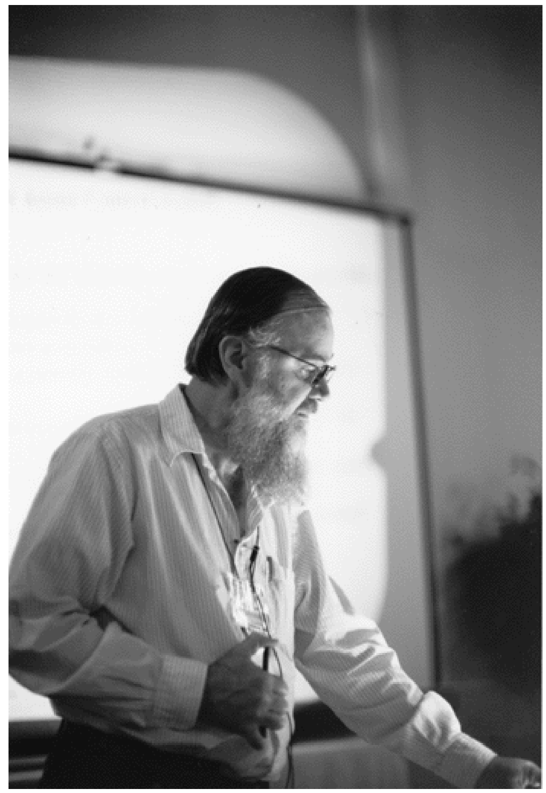
The team also plans to adapt the simulation to heterogeneous, rather than homogeneous, computers, to determine whether the deferred method works for other classes of aerodynamic simulations, and to try similar experiments using Legion, another approach to managing distributed resources. "Most single applications on the IPG will undoubtedly be run on single computers, and Globus will serve as a useful tool for getting Grid access," says Van der Wijngaart. "But our work has the promise of paving the way for more demanding applications, and of spinning off many technologies useful for mainstream Grid computing."

For more information about this research, send email to Rupak Biswas at rbiswas@nas.nasa.gov

NASnews
Main Menu
March – April 1999 • Volume 4, Number 2

To Johnston, the Grid Means Access

by Wade Roush



Last year, Ames officials invited Bill Johnston, a distinguished computer scientist from Lawrence Berkeley National Laboratory (LBNL), to lead the development of the Information Power Grid (IPG). Johnston agreed, and fortunately LBNL agreed to "lend" him to NASA for four-and-a-half days each week while the IPG effort gets underway. NAS News managing editor Wade Roush interviewed Johnston in December about his past work, his philosophy of computing, and his tactical plans for the IPG. *Photo by Tom Mead*

For many, the term "distributed computing" means dividing up a large job, such as a numerical simulation, so that it can run in concert on neighboring or distant computers. As other stories in this issue illustrate, this kind of computing will be one application of the Information Power Grid (IPG), the nationwide computing infrastructure that NASA is helping to build to support the goals of its Information Technology and High Performance Computing and Communications Programs. But ask IPG lead Bill Johnston how he understands distributed computing, and you'll get a much more sweeping view--one in which access to information, rather than aggregating computers, is the highest goal.

"I almost hesitate to use the term distributed computing, because what grids are fundamentally about is the interconnection of computer-mediated resources--and increasingly that means everything," says Johnston. Wind tunnels, telescopes, particle accelerators, medical imaging devices, immersive displays, mass storage systems, and, of course, supercomputers are a few of the scientific resources that Johnston believes future networks should link together. The goal: making it easier than ever before for scientists and engineers to access, organize, analyze, and present information about the physical world.

Johnston's extensive experience in computer graphics, security, and high-speed, wide-area, data-intensive computing has given him the broad vision that Ames officials tapped when they asked him to lead the IPG effort. In 1991, Johnston coordinated one of the first demonstrations of a large distributed application running over high-speed Internet links--a real-time remote visualization of a high-resolution MRI scan of a human brain using supercomputers. He was a co-chair of BAGNet, a high-speed networking testbed that operated from 1994 to 1996 and connected 15 Bay Area organizations from Stanford University and NASA Ames Research Center to Pacific Bell, Hewlett-Packard, and Kaiser Permanente. Johnston also leads a computer security group at Lawrence Berkeley National Laboratory (LBNL).

"I looked at what NASA wanted to do with IPG and I said, 'There is no better opportunity to take what we've learned over the last 10 years and incorporate it into a main-line computing environment,'" he told NAS News managing editor Wade Roush in a recent interview.

Roush: Tell me how you first started to think about all the technical challenges involved in high-speed, data-intensive distributed computing.

Johnston: The answer is in the things I worked on before 1988, mainly imaging systems. In 1971 I went to Lawrence Livermore National Laboratory and worked on a big coupled meteorological and air-chemistry code. It was a complete air-quality model for the San Francisco Bay Area. I got very interested in computer graphics, because this thing put out huge volumes of data, and the question was, how do you represent that data in a way that makes any sense?

Through that interest, I ended up moving from Livermore to LBNL, where I ran the computer graphics group from the mid-'70s to the mid-

'80s. It was getting to the point then where scientific instruments would routinely generate images, but the computing environment at that time was not very adept at dealing with them. The images were almost always captured on one machine and needed to be displayed on another. The whole problem of having to transport and analyze these images across more than one machine led to an increasing interest in distributed computing technology.

Roush: What form did that interest take?

Johnston: Tom Buddinger, who ran the research medicine group at LBNL, was the co-inventor of magnetic resonance imaging, and he's always been interested in ultra-high-resolution MRI. Starting around 1989 we were helping him take MRI data and convert them into 3-D datasets that could then be manipulated and viewed with such things as contrast enhancement. At that time, this could only be done on a supercomputer. And we asked, 'Well, can we do a distributed demonstration?'

It took two years to get there but the answer was, yes. We split the visualization work across the Thinking Machines CM-2 and the CRAY Y-MP at Pittsburgh Supercomputing Center, and the National Science Foundation put an NSFnet node in Albuquerque so we could do high-speed remote visualization in real time at the Supercomputing conference in 1991. That was the first year anybody had thought about running a high-speed line into Supercomputing.

Roush: Later you became an organizer of the BAGNet consortium. What did you learn from that effort?

Johnston: We learned that at that point in time it was very difficult to make distributed applications work. The thrust of BAGnet was as a multimedia testbed, and we did some very interesting things, but it was at a time in technology development when, for instance, Sun's version of [video transmission protocols] and DEC's version and HP's version did not interoperate. So I would say, by the original metric of what we set out to do, BAGNet was not a big success.

But a number of very successful things were done among the BAGNet partners. From my point of view, the most successful thing that came out of BAGNet was that we were able to put a relatively high-data-rate angiography instrument in a Kaiser hospital in San Francisco online.

We essentially set up a real-time digital library system to collect the output from this instrument in Berkeley, catalog it, process it, and make it available to Kaiser physicians in two other hospitals that were connected over BAGNet. This gave them immediate feedback about the urgency of their cases. That system embodied an awful lot of the characteristics of the IPG, particularly the remote user interfaces and the remote data management and analysis components.

Roush: What's the underlying need for that kind of high-data-rate networking?

Johnston: There are so many disciplines in which the breadth of what you can do is completely limited by the different types of data that you can get your hands on. I think science and engineering environments are really transitioning from being compute-driven to being data- and information-driven. Information environments are now being completely computer-mediated, which is fundamentally different than the situation five or six years ago.

It's probably only due to the Second World War that computers are called computers at all, as opposed to 'informators.' After the tremendous boost that numerical simulation got during the war, computers were primarily seen as computational engines. That's what's changing back now. It's obvious to anybody who uses the Web that computers are primarily mechanisms for organizing and presenting information.

I came to NASA because I believe that the grid approach to computing and data--of which IPG is an example--has the potential to make a fundamental change in how we do science and engineering R&D. NASA's commitment to IPG as the way to do computing puts them at the forefront of this revolution.

Roush: In the current draft of the IPG Implementation Plan (see NAS News, September-October '98), you break down the technology goals into basic functions like locating resources, naming and characterizing resources, and coordinating their utilization. Which of those problems do you think researchers have already got a pretty good start on, and where will the really hard work have to be done?

Johnston: The core of the document, which is changing as we speak, describes the services we want to put together in a two-year time frame-

-the things we have named Programming Services, Grid Common Runtime Services, and Resource Management. In the Programming Services area, you want to provide support for multiple programming paradigms. What do we have for paradigms right now? The answer is that most numerical codes are evolving toward FORTRAN plus MPI (the Message Passing Interface). That's exactly the environment that the Globus metacomputing toolkit [developed at Argonne National Laboratory and the University of Southern California; see story, this issue] was designed to address. At the same time, the object-oriented approach of Legion and the commercial support of CORBA make it essential to support those programming approaches as well. And the shared memory and threads paradigm commonly used on systems like the SGI Origin2000s can also many use of many Grid/IPG services.

If you're going to combine multiple resources to do a single job, then you need the sort of temporal scheduling we haven't done before. Co-scheduling involves several components. One component is the brokerage function, where users say, 'Here are all the things I would like to do, please find me resources that (A) can do them, and (B) are all available at the same time to do them.' The problem of identifying what resources are needed is pretty much done by hand now. We need to know how to automatically characterize codes so you can give a better description of what kind of resources they need. This is the topic of some university R&D that IPG is funding.

Another component is the advanced reservation problem. Right now, the job scheduler on a supercomputer may switch your scheduling if it comes up with a slot that looks appropriate. What we have to do now is put constraints on that--in other words, 'I have to have the machine between this time and that time.' You also have to be able to reserve network bandwidth. This is an R&D area where I'm actively promoting collaboration between NASA and the Department of Energy (LBNL and Argonne). Until very recently there weren't even the beginnings of the underlying mechanisms for doing that.

Roush: What are some of your criteria for the success of IPG?

Johnston: One criteria of success is that in the two-year time frame we have CoSMO [NASA's Consolidated Supercomputing Management Office] using Grid technology for some of its production supercomputing environment. We're going to work very hard to see that it happens.

Also, lots of supercomputer center users still need better password protection. Grid security intends to extend way beyond current practices to provide single login, strong authentication, and confidential communications.

I think that we'll also have a much better handle on high-performance remote data access in two years. There's no reason why researchers anywhere in the country who have a legitimate interest in wind tunnel datasets, for example, should not have routine and easy access to them.

I think what the Grid is fundamentally about is collaboration and sharing of resources. And I don't mean just computing resources, but also things like making major data archives and major instrument systems available to our collaborators around the world.

Updated: August 3, 1999
WebWork: *Publications Media Group*
NASA Responsible Official: *Bill Feiereisen*

QUESTIONS

HELP

NASnews

Main Menu March – April 1999 • Volume 4, Number 2

In this issue...

Behind the Curtain at NAS User Services

by Wade Roush

Within This Article...

Users who are new to the supercomputing and mass storage systems at the NAS Facility might be surprised at how many troubleshooters, field engineers, and computer scientists it takes to keep these machines online, respond to glitches, help customers make the best use of their allotted computing time, and maintain the NAS Systems Division's technological and administrative infrastructure.

When these professionals do their jobs well, they remain unobtrusive and largely uncredited -- but when the machines turn uncooperative, they often take the heat. To acquaint readers with the complex organization on the other side of the remote login command, this article presents an overview of the division's user support groups.

Handling user problems, questions, and requests of all kinds is the main responsibility of the NAS technical support staff. Users can reach the control room 24 hours a day, 7 days a week at 800-331-USER or 650-604-4444. To request changes to their personal information on NAS systems, users should send e-mail to **accounts@nas.nasa.gov**. Information about system status, scheduled system activity, training and publications, and more is available online.

The first group to meet: Control Room

(Photos by Judy Conlon, Wade Roush, and Dan Depauk)

NASnews
▶ Main Menu
March – April 1999 • Volume 4, Number 2

Reading Cray Binary Files on the SGI Origin2000

by Johnny Chang and Dave Dillon

Many users of Von Neumann, Eagle, and Newton, the Cray supercomputers at the NAS Facility, now have accounts on the SGI Origin2000 machines. Since the Crays and Origin2000s have vastly different hardware architectures and the default binary files generated by Fortran programs on these two systems are not interchangeable, porting codes and data from one architecture to the other becomes a non-trivial problem. This article addresses the question of porting or, more precisely, "reading" Cray (Fortran) binary data files on the Origin2000s.

Users have two major choices. They can either convert Cray binary files to SGI binary files by running a program on the Cray, or they can do the conversion on the Origin2000. Doing the conversion on the Cray is fairly straightforward. The user writes a code to read the Cray binary data file and to write the data back out (in SGI format) to another file, say SGI_file. The *assign* command is used prior to running the executable to specify the form of the output SGI_file.

Two crucial options to the *assign* command are the *-F f77* and *-N ieee_64* options. For example, the command:

```
cray_prompt%  assign -a SGI_file -F f77 -N
ieee_64 u:11
```

will cause the output from the user's code (assuming it is written to unit 11) to go through the appropriate I/O library layers and to write to an output file (SGI_file in this example) with FORTRAN 77/UNIX Fortran record blocking (default structure for sequential unformatted files on IRIX systems) and 64-bit IEEE numeric representation (64-bit reals and 64-bit integers). Other appropriate arguments to the *-N* option for the Cray to SGI conversion are *ieee_dp* and *ieee* (or *ieee_32*). The former gives 64-bit floating point numbers and 32-bit integers while the latter gives both reals and integers in 32-bit representation. If the loss in precision of the latter is tolerable, then one does not need to write a conversion program since an *assign* command on the SGI will already do the job.

File conversion is easy on the Cray, but there are many reasons why a user might prefer to do it on an Origin2000 instead. Users may simply want to avoid using their Cray time allocation to do file conversion, for example, or they may not know exactly which old files will be needed in the future. The Origin2000 IRIX system also has an *assign* command with a subset of the features that have been implemented on the Crays. At present, with the current generation of the MIPSpro 7.2 Fortran compiler, it is possible to convert Cray binary data to IEEE 32-bit data "on the fly," that is, as it is read into the user's application. The *assign* command on the Origin2000 that accomplishes this for a file called Cray_file connected to unit 12 is:

```
O2k_prompt%  assign -a Cray_file -s cos -N
cray u:12
```

The *-s cos* option specifies that the file uses cos blocking (default structure for sequential unformatted files on UNICOS systems). The *-N cray* option tells the I/O library that the floating-point numbers in the binary file are in Cray's numeric representation. Note: It is necessary to set an environment variable FILENV to some file for holding all the input file attributes specified by the *assign* command. For C-shell, one uses:

```
O2k_prompt%  setenv FILENV assignfile
```

prior to invoking any *assign* commands. The *assign -V* option shows which assign attributes have been set, and *assign -R* clears all assign attributes.

To use this method, all the variables/arrays in the code that will hold Cray binary data must be declared with default length (32-bit), and no options that change their length (that is, *-r8* or*-i8*) can be used during compilation. Furthermore, if the f77 compiler is used, *-craylibs* must be appended to the end of the compile/link command line.

Unfortunately, this 64-bit Cray to 32-bit IEEE truncation can cause an unacceptable loss in precision, and, for long integers, the conversion is simply wrong. (More precisely, large positive integers greater than or equal to 2**31 "wrap around" to negative integers.) It is anticipated that a future release of the MIPSpro Fortran compiler (version 7.3) will allow 64-bit Cray to 64-bit IEEE data conversion using an *assign* command.

Currently, to obtain 64-bit Cray to 64-bit IEEE conversion on the Origin2000, one must use a combination of the *assign* command to specify the cos record blocking and a CRY2MIPS function call within the user's application for converting the numeric representation of the data. The CRY2MIPS function must be called for converting each variable/array that will be holding floating-point numbers. All integers and floating-point real numbers must be declared with a byte length of 8, and complex numbers with a byte length of 16. The compilation of the user program must not contain any options that change the length of the variables inside the program. The following example for converting a mixture of character data, integers, and real and complex numbers illustrates these points.

The first program runs on a Cray and creates a Cray binary file called fort.7.

```
cray_prompt%  cat cray_data.f
program cray_data
character*8 name1, name2
integer i(4)
real a(4)
complex z(2), ci

pi = acos(-1.0)
ci = (0.0,1.0)
name1 = 'hi there'
name2 = 'abcdefgh'

a(1) = pi
a(2) = 1.23456789e+10
a(3) = 9.87654321e-20
a(4) = -1.23456789e+30

i(1) = 123456
i(2) = 123456789
i(3) = 123456789012
```

```
i(4) = 123456789012345

z(1) = a(1) + ci*i(1)
z(2) = a(2) + ci*i(2)

write(7) name1, a(1), a(2), i(1), i(2),
z(1)
write(7) name2, a(3), a(4), i(3), i(4),
z(2)
end
```

The fort.7 binary file is ported to an Origin2000 and is read by the following program. Note that the lengths of the integer, real, and complex data to be read in are explicitly declared within the program. Note also that it is not necessary to call CRY2MIPS to convert integer and character data. See the CRY2MIPS manpage for more details on its usage.

```
O2k_prompt%   cat read_cray.f
program read_cray
character*8 name1, name2
integer*8 i(4)
real*8 a(4)
complex*16 z(2)

read(7) name1, a(1), a(2), i(1), i(2), z(1)
read(7) name2, a(3), a(4), i(3), i(4), z(2)

ierr1  = cry2mips(3, 4, a(1), 0, a(1), 1,
64, 64)
ierr2  = cry2mips(4, 2, z(1), 0, z(1), 1,
128, 128)

print *, 'name1 = ', name1
print *, 'name2 = ', name2
print *

print *, 'pi   = ', acos(-1.0)
print *, 'a(1) = ', a(1)
print *, 'a(2) = ', a(2)
print *, 'a(3) = ', a(3)
print *, 'a(4) = ', a(4)

print * print *, 'i(1) = ', i(1)
```

```
print *, 'i(2) = ', i(2)
print *, 'i(3) = ', i(3)
print *, 'i(4) = ', i(4)

print *
print *, 'z(1) = ', z(1)
print *, 'z(2) = ', z(2)
end
```

The read_cray.f program is compiled without any variable length adjusting options and is run under the control of the following cos blocking *assign* attribute:

```
O2k_prompt%   f90 -o read_cray read_cray.f
 (append -craylibs when using the f77


compiler)
O2k_prompt%   setenv FILENV .assign
O2k_prompt%   assign -s cos u:7
O2k_prompt%   ./read_cray name1 = hi there
name2 = abcdefgh

pi   =  3.14159274
a(1) =  3.1415926535897967
a(2) =  12345678900.
a(3) =  9.8765432099998414E-20
a(4) =  -1.2345678900000221E+30

i(1) =  123456
i(2) =  123456789
i(3) =  123456789012
i(4) =  123456789012345

z(1) =  (3.1415926535897967,123456.)
z(2) =  (12345678900.,123456789.)
```

The output shows pi to be different from a(1) because of differences in their precision. This illustrates an important point: If codes to be run on an Origin2000 depend on compiler options to set their variables to double precision, then it is better to write a separate stand-alone code for file conversion rather than make extensive changes in the user's application.

*For more help with data conversion, contact the NAS scientific consulting group at (650) 604-4444 or (1-800) 331-8738, or send email to **nashelp@nas.nasa.gov**.*

NASnews
▶ Main Menu
March – April 1999 • Volume 4, Number 2

NAS Heightens Computer Security

by Alex Woo

When the NAS Facility was created in 1986, aerospace computation was primarily academic, and the Internet explosion had not yet begun. Today, advanced, sensitive, or proprietary technology is part of nearly every simulation carried out in the NAS Systems Division, and the nature and extent of the Internet have vastly changed. So, too, has the threat to NAS computer security. All NAS system users should be aware of the threat and of measures to counteract it.

In December alone, approximately 90 sustained probes or attacks were reported to the NAS computer security team.

In 1998, NAS detected three root-level security breaches, including the compromise of root passwords on all major systems and password collection on our gateway server. In December alone, approximately 90 sustained probes or attacks were reported to NAS computer security, about 50 of them from the U.S. The overwhelming majority of these attempts were mere nuisances perpetrated by "script kiddies." However, owners of NAS computer files should assume that some are targeting sensitive U.S. technology.

Security vulnerabilities at NAS and other computing facilities can be loosely divided into three areas: internal network threats, external network threats, and compromises from within. For traffic on internal networks, some existing Unix security and authentication schemes trust the integrity of remote machines according to their port number or IP response. But in today's world, remote machines can be compromised at any time, and it should not be assumed that a plausible physical location or IP address guarantees integrity. On its internal systems, NAS is transitioning from freeware Unix security and configuration checkers such as "tripwire" and "cops" to the commercial ISS System Configuration Scanner and TIGER.

The facility is also aware that Ethernet, FDDI, Token Ring, and other internal networks are essentially party lines. If any system on the line is compromised, all of the traffic can be captured. To reduce its vulnerability, NAS is converting internally to switched Ethernet, and is continuing to evaluate Secure Shell (SSH) as a way to create private, encrypted virtual networks.

To counteract external network threats, NAS has been scanning its systems with the network penetration tool ISS Internet Scanner for over a year. The facility is also cracking password files each October. For a year, NAS has also been using ISS Real-Secure intrusion detection software, which has recently been deployed on the FDDI backbone. 100 percent of NAS's network traffic is now monitored, and in 1998 NAS completed system sensitivity and risk assessments for all hosts in the division.

Attacks on computer systems from within can be the most damaging, since a well-placed insider can pinpoint key data quickly. None of the operating systems used at NAS are immune to root compromise from an existing user account. A fair percentage of the exploits in the hacker repositories worked when tested on SGI, IRIX, Linux, and SunOS systems at NAS.

The moral: Sensitive data should be encrypted on NAS filesystems. All NAS users are required to sign an "acceptable use" policy stating that they are responsible for protecting information used or stored in their accounts. Users with sensitive or export-controlled data should encrypt it whenever it resides on a disk, tape, or optical medium. If one account is compromised, it endangers all the other systems on the network, so users should employ SSH or other session encryption packages or one-time password or challenge-response systems to protect sensitive information such as passwords and passphrases.

The high-performance computing community is hard at work on a standard authentication and security infrastructure. The National Computational Science Alliance, the National Partnership for Advanced Computational Infrastructure (NPACI), and the NAS-led Information Power Grid project (IPG) are contributing to this infrastructure. Alliance, NPACI, and Department of Defense sites have already implemented a combination of Kerberos, SSH, and challenge-response authentication systems, and all have disabled clear-text passwords. When a standard is established, NAS will adopt best-practices and disable clear-text passwords on untrusted networks.

Public key cryptography based on trusted certificates is one future direction for NAS and the IPG. Globus security services, as used on the IPG research testbed, are based on Secure Socket Layer x509v3 certificates. Versions of SSH and ftp using Globus certificates have been written, and in the future, these packages may also accept NASA and Federal PKI certificates. These emerging standards and protocols will assist NAS in strengthening its own security infrastructure, while also assuring a smooth integration with the high-performance computing community.

For more information about computer security at NAS, see the Security Team website.

NASnews

▶ Main Menu

March – April 1999 • Volume 4, Number 2

Credits

Executive Editor: Bill Feiereisen

Managing Editor: Wade Roush

Contributing Writers: Johnny Chang, Dave Dillon, Chris Henze, Alex Woo

Photographers: Judy Conlon, Michael Boswell, Dan DePauk, Tom Mead

Online Page Layout and Graphics:Joel Antipuesto, Chinley Chang, Eunah Choi, Jill Dunbar, Chris Gong, Wade Roush

Special thanks to: Richard Anderson, Stephen Barnard, Rupak Biswas, Pieter Buning, MiYoung Cho, Po Chung, Christine Cortez, Chrystal Cowdrey, Lisa Coyle, James Donald, Jill Dunbar, Peggy Fenner, Chris Henze, Mary Hultquist, William Johnston, Randy Kaemmerer, David Kenwright, Tim Kirby, Paul Lambert, Daniel Mercer, Patrick Moran, George Myers, Chuck Niggley, Todd Novak, Alan Powers, Tom Preece, Marcia Redmond, Darrell Root, Han-Wei Shen, Eugene Tu, William Van Dalsem, Rob Van der Wijngaart, Chris Williams, Andy Wissink, Maurice Yarrow, John Ziebarth

Lawrence Livermore National Laboratory ensures national security and applies science and technology to important problems of our time.

Privacy & Legal Notice

American flag

Security Update

News

Lab Scientists Present Cosmic Findings to Astronomical Society

Scientists Create Virtual Star Over Hawaii

Four LLNL Scientists Named APS Fellows

Search

Go

Find People

Go

Index | Help | Comments

Safety

Jobs

Librar

National Nuclear

News & Public Affairs

Working with LLNL

Science & Technology

Education

A correlated sensor network being developed at Livermore can help fight nuclear terrorism and other threats.

<-- More Info -->

Current issue of Science & Technology Review

Publications

Organization

About the Lab

NASnews

Main Menu

March – April 1999 • Volume 4, Number 2

In this issue...

High-speed processor group. L to R: Dave Dillon (SGI), Matt Cary, Scott Emery (SGI), Bill Plummer (SGI), Alan Powers (group lead), Bron Nelson (SGI), Tim Kirby, Eric Langhirt (control room), Sarita Wood. (Photo by Judy Conlon)

High-speed Processor Group

While the control room keeps the super-computers running and the LAN group keeps them connected, it's the members of the high-speed processor (HSP) group who ensure that customers can use them for production work. The CRAY C90s and J90 managed by the NASA Consolidated Supercomputing Management Office (CoSMO) and the seven Silicon Graphics (SGI) Origin2000 machines owned by NASA's Data Assimilation Office are the facility's main production machines, which are supported 24 hours a day, 7 days a week. "We change and install operating systems and other software on these systems, do testing to make sure the equipment will work under stress, and diagnose problems to determine whether they're in the hardware or the software," says Alan Powers, the HSP Group lead. "If it's the hardware, the field engineers take over. For system problems, we're the first line of defense."

Urgent problems, such as failed login attempts or compiler bugs that prevent programs from loading, have first claim on the HSP group's attention. More often, they help users who have exceeded their disk quota or whose jobs seem to be languishing in queues. Whatever and whenever the problem, the P.O.C. carrying the group's pager and cell phone fields the call. "Sometimes the P.O.C. gets awakened two or three times a night," says Powers.

Next: Parallel Systems/Metacenter

Back to introduction

NASnews

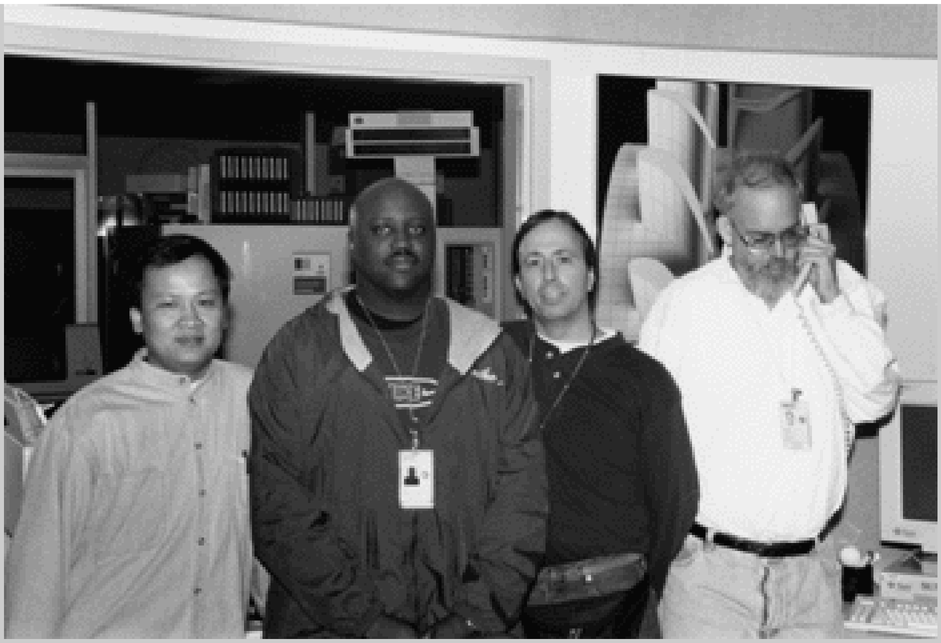
▶ Main Menu   March – April 1999 • Volume 4, Number 2

In this issue...

Control room, 1st shift. L to R: Ron Pesta, Chris Grimes, Frank Sabadin (shift lead), Sam Thomas, Eric Langhirt. (Photo by Dan DePauk)

The Control Room

Three shifts of analysts work around the clock in the control room, the "triage" center of the NAS Facility. When a remote or local problem requires attention, the control room analysts are the user's first point of contact. They answer the NAS Help Line and log each caller's contact information and problem as a new "ticket" in Remedy, the call management program used by NAS to track user problems and requests.



Control room, 2nd shift. L to R: Thomas Nguyen, Terry Brown (shift lead), Ernst Kimler, Ray Turney. (Photo by Dan DePauk)

Then their knowledge, experience, and discretion come into play. The analysts must decide whether a problem is something they can fix, by translating an error message, for example, or rebooting a machine. If not, they must decide which of the specialized technical groups can best handle the problem. "A lot depends on the skill level of the analyst who takes the call," says Chrystal Cowdrey, a former control room staffer who now helps maintain the Remedy system. "If they know Fortran, for example, they can read an error message and tell that the problem is in the source code, not the machine. So they would send the ticket to the scientific consulting group."

Control room, 3rd shift. Bill Arasin, Rich Cooter. Not pictured: Richard Paynter, Michael Stitzel, Gabe Wedekind. (Photo by Dan DePauk)

The control room analysts initiate about 1,000 Remedy tickets each month. They are more than just help-line operators, however. Each is responsible for monitoring specific machines on the computer room floor, adjacent to the control room. They log into the machines and periodically examine batch queues, file systems, CPU utilization, and other important measures of system health. (Most of these indicators are also monitored continuously by an in-house system called Centralized Testing and Management Software, which reports unusual events.) The analysts also take a walk around the computer room every hour or two to check for console error messages and tape problems, detect smoke from hidden electrical shorts, and so forth. "Sitting around and waiting for calls doesn't happen a lot," says Cowdrey.

Next: LAN Group

Back to introduction

NASnews
▶ Main Menu
March – April 1999 • Volume 4, Number 2

In this issue...

Local area network group. L to R: Joseph Soleido, Dan Borlean, Andrew Flury, Steve Fuqua, Steve Tragoutt, Eddie Ordaz, Greg Smith, Darrell Root (group lead), Chris Williams. (Photo by Judy Conlon)

LAN Group

Ensuring that the supercomputers can talk to each other, to the hundreds of workstations at the NAS Facility, and to the outside world is the job of the local area networking (LAN) group. "We're responsible for everything on this side of fix1 and fix2," routers that connect the NAS LAN to the Ames network and the Internet, explains group lead Darrell Root. That means high-speed HIPPI channels between the supercomputers, FDDI channels to the fixed router, and 10- or 100-megabit-per-second Ethernet links to nearly all of the building's workstations.

The LAN group's main challenge is to support this variety of networks, serving an even greater variety of hosts, Root says. The machines connected to the network include Cray, Convex, and Silicon Graphics high-speed processors, Sun and IRIX workstations, and PCs running Macintosh, Windows, and Linux operating systems. Managing constant additions and changes to the network is a job in itself. "If we have to make a configuration change to one machine on the HIPPI net, for example, we need to manually update the address information on every other machine," Root says. "Or if Boeing changes their network topology, our routing configuration can get out of whack with their network, and a correction needs to be made."

When not acting as the designated "point of contact" (P.O.C.) for the day, LAN group members respond to networking problems reported to the control room. Otherwise, they work on long-term projects such as the ongoing rewiring of the main NAS building. "We're removing old cable that isn't up to new fire codes, upgrading copper wiring to support Fast Ethernet, and installing fiber connectivity to the desktop for gigabit Ethernet, HIPPI, and video," Root explains.

Next: High-speed Processor Group

Back to introduction

Parallel systems group. L to R: Archie de Guzman, Bill Petray, Lou Zechtzer, Jens Petersohn, Karl Schilke, Rita Williams. Not pictured: Mary Hultquist (group lead). (Photo by Wade Roush)

Parallel Systems/Metacenter

What the HSP group does to support the production supercomputers, the parallel systems group does for the testbed machines, between 5:00 a.m. and 5:00 p.m. Pacific Time. At present, all of the testbed machines are parallel-processing, shared-memory SGI Origin2000s, connected in a configuration called the "NAS Origin Cluster." The cluster is used mainly for experiments in scaling, the parallelization of software to run on multiple processors simultaneously, and metacomputing, the joining of neighboring or distant computers into a common pool of resources. The group leads the NASA metacenter linking Ames, Langley, and John Glenn (formerly Lewis) Research Centers and consults on a metacenter linking two Department of Defense supercomputing centers.

"We're trying to get the testbeds up, stable, and workable, which gets interesting as we begin to hook the Origin2000s together into systems of 256 processors and more," says Mary Hultquist, the group's lead. "A lot of work goes into coming up with workarounds and solutions to problems inherent to beta software." The group works with vendors to identify and implement the latest software, and with the scientific consultants to test systems before release to users.

Hultquist says the group's everyday tasks, such as identifying compiler problems and investigating the causes of variations in a program's runtime behavior, are good practice when the team turns to designing new metacenters. "The fact that we're supporting a system gives us real-world experience of what users need, which we wouldn't get if we were developing new systems in a vacuum."

Next: Scientific Consulting

Back to introduction

In this issue...

Scientific consulting group. L to R: Ed Hook, Johnny Chang, Clayton Guest, Chuck Niggley (group lead), Samson Chung, Steve Heistand, Bob Hirsch, George Myers, Terry Nelson. (Photo by Judy Conlon)

Scientific Consulting

The scientific consulting group is available every weekday from 7:00 a.m. to 5:30 p.m. Pacific Time to respond to Remedy tickets and to help users iron out problems in their code or optimize performance on any of the high-performance platforms. (Because the Origin cluster is used mainly to test new computing techniques, copies of the parallel systems group's Remedy tickets automatically go to the scientific consultants as well.) All of the consultants have programming backgrounds, and many come from computer-industry giants such as Control Data Corp., Cray Research, and Intel. "We've got over 200 years of computing experience in the group," says Chuck Niggley, the group lead.

The group's goal is to close 80 percent of its assigned Remedy tickets within 48 hours. But other cases can be far more involved. When a user needs help porting software from a vector machine to a parallel machine, for example, the work can go on for weeks, says Niggley. In fact, helping users adapt to changing technology takes the majority of the group's time, he says. "Because the Origin2000s are new, the systems people are always introducing new software releases and bug fixes. It's like living on a fault line," he says. In addition to solving problems, the group also helps users maximize the amount of computing they can get done with their allocated resources. Niggley says that may include developing online training materials, contributing how-to articles to NAS News, and sometimes traveling to customer sites to teach classes.

Next: Mass Storage

Back to introduction

NASnews
► Main Menu
March – April 1999 • Volume 4, Number 2

Mass storage group. L to R: Chris Shaw, Tom Preece (group lead), Larry Spoo, Tony Luggiero. (Photo by Wade Roush)

Mass Storage

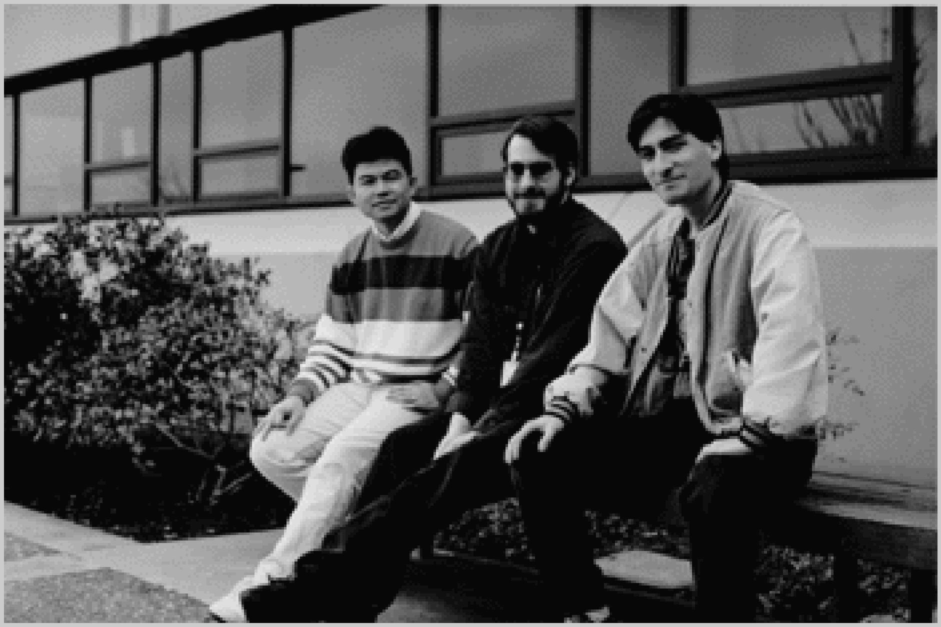
When they're not being crunched by the supercomputers, all those numbers need somewhere to stay. The NAS mass storage group makes sure that they always have a place to reside.

Chuck and Scott, the NAS Facility's mass storage devices, can store up to 2.4 terabytes of data on disk and 105 terabytes on tape. The group monitors tape pool levels inside the storage silos, ensuring that users have media to which to write, and they help restore old files or reconstruct databases after a machine crashes in the middle of a write operation. That happens more often than users or the mass storage group would like, as Chuck and Scott approach their eighth year of service. Another of the group¹s tasks, therefore, is to prepare for the changeover to a new mass storage system, expected this year. "A lot of our time is spent creating scripts for the upcoming storage system," says Tom Preece, a member of the group. "We're also streamlining and repackaging data so that when we transfer it over, it will be a lot cleaner."

Next: Distributed Software

Back to introduction

NASnews
Main Menu
March – April 1999 • Volume 4, Number 2



Distributed Software group. L to R: Po Chung, Andy Meyer, Ron Echeverri. Not pictured: Phil Kleiber. (Photo by Judy Conlon)

Distributed Software

Groups with whom remote users don't often cross paths, but whose work is crucial to the facility's operation, include the distributed software group, the database group, the workstation group, and the Macintosh support group. The first, as its name implies, is responsible for installing and upgrading software used on a variety of computing platforms. This includes both specialized visualization codes such as UFAT and general purpose programs such as the GNU utilities.

"Users expect the same software whether they're logged onto the Crays, the Origin2000s, or a workstation," explains group lead Po Chung. The group also manages the email accounts provided to all NAS users, runs the Ames-wide Usenet news server, and maintains the division¹s 40-plus computer printers.

Next: Database Group

Back to introduction

NASnews
▶ Main Menu
March – April 1999 • Volume 4, Number 2

In this issue...

Database group. L to R: MiYoung Cho, Cassandra Baldassano. Not pictured: Chrystal Cowdrey, Cindy Ruan. (Photo by Judy Conlon)

Database Group

The database group manages Acct++, a database that collects, tracks, and reports data on the utilization of many systems at NAS. The group ensures that the client programs collecting this data run correctly, and that billing programs are charging resources consumed by each job to the proper user or project. Using in-house software tools, the group can help users access data on their computer usage (by user, project, and machine).

The LAMS database, which administers information for all user accounts and groups on more than 400 Unix-based systems controlled by the NAS Systems Division, is also managed by the database group. The database controls information for all user accounts, groups, and machines. While the control room staff creates, modifies, and deletes accounts using LAMS, the database group maintains LAMS by fixing bugs, making upgrades, and installing the client-server program that provides encrypted communications between LAMS and its supported systems. The group also maintains the Remedy system and the SysInfo database used by the workstation, Mac Support, and facility staffs to track equipment owned and supported by the division.

Next: Workstation and Macintosh Support

Back to introduction

NASnews
▶ Main Menu
March – April 1999 • Volume 4, Number 2

In this issue...

Workstation Group. L to R: Peggy Fenner, Frank Cianci, Lisa Yeung, Daniel Sully, Deryl Castellano, Sabrina Farmer, Sophia Mai, Don George. (Photo by Judy Conlon)

Workstations and Macintosh Support

Both support personnel and researchers at NAS need substantial computing power on their own desktops. The workstation team maintains almost 500 machines at the NAS Facility, including about 320 Unix-based SGI workstations, 80 Sun workstations, and 100 PCs running Windows and Linux. The Macintosh support group cares for another 350 machines, most of them powerful G3s.

Macintosh support group. L to R: Christine Cortez (group lead), Todd Novak.. (Photo by Judy Conlon)

Both groups handle ongoing jobs such as software installation and upgrades, security, file server and backup server maintenance, and troubleshooting. Their first job every morning, however, is to check for workstations or Macs that are out of commission. "We try to catch these machines proactively, so users don¹t find them down when they come in," says Peggy Fenner, a workstation team member.

When the user finds a problem first, however, "It's our job to answer the call, see if it¹s something that can be handled over the phone, and if not, do some hands-on fixes," says Mac support group lead Christine Cortez.

Back to introduction